REMARKS

Reconsideration and further examination are respectfully requested. Claims 1-3 and 5-18 are currently pending.

Reconsideration and further examination are respectfully requested. Claims 1-3 and 5-18 are currently pending.

Claims 1-3, 5 & 17:

Rejections under 35 U.S.C. §103

Claims 1-3, 5 and 17 were rejected under 35 U.S.C. §103(a) as being unpatentable over Davis et al. (U.S. 5,357,617) in view of Epps et al. (U.S. 6,813,243)..

Davis:

Davis describes, in the Abstract:

"... hybrid pipelined processor and associated processing methods ... for separately handling substantially concurrently in a time division manner multiple ***program instruction threads.*** The hybrid architecture includes an instruction fetch unit, an instruction decode unit and an execution unit. The execution unit includes multiple sets of register files each of which contains the working contents for a corresponding one of a plurality n of instruction threads. Timing and control circuitry is coupled to each of the principal processor components for controlling the timing and sequence of operations on instructions from the plurality n of instruction threads such that multiple instruction threads are separately handled substantially concurrently...."

At column 2, lines 47-48 Davis describes "... each of the instruction threads is defined by a series of instructions contained within instruction memory associated with the pipelined processor...The pipelined processor includes n register file sets, each of the n register file sets contain the working contents for a corresponding one of the plurality of n instruction threads'

At column 4, lines 36-38, Davis describes "... If desired, program instructions could be compiled into three or more instruction threads for processing by a single pipelined processor pursuant to the present invention.... "

Epps:

Epps describes, in the Abstract:

"... A pipelined linecard architecture for receiving, modifying, switching, buffering, queuing and dequeuing packets for transmission in a communications network. The linecard has two paths: the receive path, which carries packets into the switch device from the network, and the transmit path, which carries packets from the switch to the network. In the receive path, received packets are processed and switched in an asynchronous, multi-stage pipeline utilizing programmable data structures for fast table lookup and linked list traversal. The pipelined switch operates on several packets in parallel while determining each packet's routing destination. Once that determination is made, each packet is modified to contain new routing information as well as additional header data to help speed it through the switch. Each packet is then buffered and enqueued for transmission over the switching fabric to the linecard attached to the proper destination port. .."

Epps describes, in the Abstract "... Both enqueuing and dequeuing of packets is accomplished using CoS-based decision making apparatus and congestion avoidance and dequeue management hardware..."

The Examiner states, at page 5 of the office action:

"... it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine Davis' ideas of using the pipelined processor for substantially concurrent processing of a plurality of program instruction threads simultaneously into Epp's pipelined switch/linecard which parallel operates on a number of IP packets for transmitting them over IP/ or TCP communication network in order to be able to employ a well-known standard into Epp's system for saving resources and development time to bring up at least one advantage i.e., provide more flexibilities for IP network switch system, see (Epps: column 2, lines 7-65; column 8, lines 1-8).

Applicants Argument

As described at pages 4-6 of Applicant's specification, as well as in the claims as shown above, Applicant's invention is directed at multi-stage packet processing pipeline, where each

stage is multi-threaded. Each processing stage has the ability to process a large number of packets. Each multi-IP packet thread and processing pipeline stage are independent, and multi-IP packet threads are forwarded as a unit between the processing stages, thereby allowing large numbers of packets to be processed by each pipeline stage before transferring the multi-IP packet thread to a following stage.

Applicants have amended the claims to more particularly highlight the architecture of the present invention, and in particular have included the limitation of the activity status that is maintained with each thread, and used to signal that the thread is able to be pipelined by the next pipeline stage.

The 'threads' and pipeline architecture of the present invention differ significantly from the program instruction threads of Davis. In the pipeline of Davis:

"... A timing and control circuit is coupled to each of the instruction fetch unit, instruction decode unit, and execution unit for controlling the timing and sequence of operations on instructions of the plurality n of instruction threads such that multiple instruction threads are separately handled substantially concurrently in a time division manner, with the hybrid pipelined processor switching between instructions of the plurality n of instruction threads..."

Thus, although Davis describes a multi-instruction thread, different instructions in the threads are being executed by different stages of the pipeline at any given time. Applicants have revised any ambiguities in claim 1 to have the claim conform more closely to claim 5 as previously presented, which claimed "...each of the plurality of stages of the processing pipeline is operating on a different one of the multi-IP packet threads..." Thus the architecture of the present invention is fundamentally different from that of Davis. Epps describes a traditional pipelined architecture that includes a fetch stage, pre-process stage, pointer lookup stage, table

lookup stage, etc. As described at column 6, lines 3=7 "... Each pipeline stage works on a different packet header portion. When the operations of all stages are complete, each stage passes its results on to the next stage at the same time..." Thus the present invention, in which each pipeline stage processes a multi-IP packet thread before passing it to the next stage, is clearly distinguished from Epps as well.

Accordingly, for at least the reason that the combination of Epps and Davis fails to describe or suggest the limitations of claim 1, including the newly added limitation regarding the maintenance of activity status for each multi-IP packet thread, it is requested that the rejection be withdrawn. Claims 2 and 3 serve to further limit claim 1 and are allowable for at least the same reasons as claim 1.

Independent claim 5 is patentably distinct over the combination of Epps and Davis, which fails to describe or suggest "...a processing pipeline including a plurality of stages coupled to receive and process the plurality of independent multi-IP packet threads such that, during a processing period, each of the plurality of stages of the processing pipeline is operating on a different one of the multi-IP packet threads..." Accordingly, for at least the reason that neither Epps nor Davis, alone or in combination describe a pipeline processing stage that processes a multi-IP packet thread, the claim is patentably distinct over the references. It is therefore requested that the rejection be withdrawn. Dependent claim 17 serves to further limit claim 5 and is therefore allowable with claim 5.

Claims 6-16 and 18:

Claims 6-16 & 18 were rejected under 35 U.S.C. §103(a) as being unpatentable over Berenbaum-Davis in view of Epps and further in view of Eickemeyer.

Eickemeyer:

Eickemeyer describes, in the abstract:

"In a simultaneous multithread processor, a flush mechanism of a shared pipeline stage is disclosed. In the preferred embodiment, the shared pipeline stage happens to be one or all of the fetch stage, the decode stage, and/or the dispatch stage and the flush mechanism flushes instructions at the dispatch stage and earlier stages. The dispatch flush mechanism detects when an instruction of a particular thread is stalled at the dispatch stage of the pipelined processor. Subsequent instructions of that thread are flushed from all pipeline stages of the processor up to and including the dispatch stage..."

Thus, like Davis, Eickemeyer describes a particular architecture of a multi-*instruction* thread processor. Thus even if a motivation could be found, the modification still would not teach or suggest all of the limitations of the claims, since Eickemeyer teaches nothing beyond that of Davis. For at least this reason it is requested that the rejection of the claims 6-16 and 18 be withdrawn and the case be allowed to issue...

Conclusion:

Applicants have made a diligent effort to place the claims in condition for allowance. However, should there remain unresolved issues that require adverse action, it is respectfully requested that the Examiner telephone Applicants' Attorney at the number listed below so that such issues may be resolved as expeditiously as possible.

For these reasons, and in view of the above amendments, this application is now considered to be in condition for allowance and such action is earnestly solicited.

Respectfully Submitted,

July 2, 2007
Date

__/Lindsay G, McGuinness/____
Lindsay G. McGuinness, Reg. No. 38,549
Attorney/Agent for Applicant(s)
McGuinness & Manaras LLP
125 Nagog Park
Acton, MA 01720
(978) 264-6664

Docket No. 120-100
Dd: 7/2/2007